PCA no Sensoriamento Remoto

MAT235 – Álgebra Linear Numérica
Autores: Christian Rodrigues, Chrystian Melo

Introdução

Imagens de satélite multiespectrais fornecem uma rica descrição espectral da superfície terrestre ao capturar diferentes faixas do espectro eletromagnético. Embora essas informações sejam valiosas para identificar elementos como vegetação, corpos d'água e áreas urbanas, elas frequentemente apresentam alto grau de redundância — isto é, as bandas espectrais costumam estar altamente correlacionadas entre si.

A Análise de Componentes Principais (PCA) é uma técnica matemática que permite transformar essas bandas correlacionadas em um novo conjunto de bandas não correlacionadas, conhecidas como componentes principais. Cada componente representa uma direção no espaço dos dados onde a variância é máxima, permitindo que a complexidade e redundância das imagens originais sejam reduzidas, ao mesmo tempo em que as informações mais relevantes são preservadas.

Este trabalho tem como objetivo a implementação do zero do algoritmo de Análise de Componentes Principais (PCA), baseado no que vimos na disciplina de Álgebra Linear Numérica, com foco especial nos conceitos de autovalores, autovetores, mudança de base e projeção vetorial.

A aplicação foi realizada em imagens reais do satélite Landsat 9, contendo bandas espectrais do visível ao infravermelho próximo. Este projeto une ambas as técnicas, implementando-as do zero em Python e aplicando-as em cenas Landsat da região de Ibirité-MG.

Objetivos

A seguir, são apresentados os principais objetivos do trabalho, acompanhados de suas respectivas justificativas e orientações metodológicas.

1 – Compreender a fundamentação matemática da PCA como técnica de Álgebra Linear

A PCA é, em sua essência, uma transformação de base ortogonal construída a partir dos autovetores da matriz de covariância dos dados. Portanto, antes da implementação prática, o projeto dedica-se ao estudo de teoremas fundamentais da Álgebra Linear, como o Teorema Espectral, a Diagonalização de Matrizes Simétricas e a Propriedade de Ortogonalidade dos Autovetores.

2 – Implementar a PCA do zero com foco na decomposição espectral da matriz de covariância

Este objetivo visa evitar o uso de bibliotecas prontas de machine learning. O algoritmo foi codificado utilizando apenas operações vetoriais e matriciais, com base no cálculo direto da matriz de covariância e sua subsequente decomposição em autovalores/autovetores via *numpy.linalg.eigh*.

3 – Aplicar a PCA em bandas espectrais de imagens Landsat 9 para realce de feições geográficas

Após processar e vetorizar as imagens de bandas espectrais, estas são transformadas usando PCA, gerando novas imagens correspondentes aos componentes principais. Os primeiros componentes são então analisados quanto à sua capacidade de realçar estruturas geográficas relevantes, como vegetação, corpos d’água e áreas urbanizadas.

4 – Avaliar a variância explicada e justificar a seleção dos componentes principais

A partir dos autovalores obtidos, foi calculada a variância explicada por cada componente e seu valor acumulado. Apenas os componentes responsáveis por ao menos 99% da variância total foram mantidos, conforme é comum em aplicações de redução de dimensionalidade.

5 – Relacionar os resultados práticos da PCA com a teoria matemática envolvida

Por fim, buscamos interpretar os resultados visuais com base na estrutura algébrica da transformação.

A análise dos autovalores e autovetores revelou como diferentes feições são separadas nas direções principais de variância dos dados.

Fundamentação Teórica

A seguir, destacamos os principais fundamentos aplicados ao longo do projeto.

PCA via SVD (Singular Value Decomposition)

Utilizado diretamente nesta implementação, a PCA foi feita via SVD, decompondo a matriz de dados diretamente como:

$$X = U\Sigma V^\top$$

- As colunas de V (ou linhas de V^T) correspondem aos autovetores.
- Σ^2 fornece os autovalores.
- os **autovalores** da matriz de covariância são $\lambda_i = \sigma_i^2/(N-1)$

Projetando:

$$Y = X\,V_k$$

Obtêm-se os k componentes principais que preservam a maior variância possível no sentido da norma 2.

K-Means (Lloyd-Max)
O algoritmo alterna entre (i) atribuir cada amostra ao centróide mais próximo e (ii) atualizar centróides pela média das amostras atribuídas. A implementação `_snap_like_kmeans` segue o SNAP: treino em amostra aleatória (até 50 000 pixels), padronização por banda e ordenação final dos clusters por tamanho para manter coerência entre execuções.

Matriz de Covariância Σ

A matriz de covariância Σ mede o quanto cada par de variáveis (bandas espectrais) varia conjuntamente. Em notação matricial:

$$\Sigma = \frac{1}{n-1} X^\top X$$

onde X é a matriz de dados centralizada (média zero por coluna).

- Σ é uma matriz simétrica e semi-definida positiva.

- Pode ser diagonalizada com autovalores reais e autovetores ortogonais (Teorema Espectral).

Teorema Espectral para Matrizes Simétricas

O Teorema Espectral afirma que qualquer matriz simétrica real pode ser diagonalizada por uma base ortonormal de autovetores.

$$\Sigma = Q\Lambda Q^\top$$

- Q: matriz ortogonal contendo os autovetores como colunas.
- Λ: matriz diagonal com os autovalores.
- Aplicação: os vetores de Q formam a nova base ortogonal para representar os dados.

Mudança de Base Ortogonal

A projeção dos dados na nova base se dá por:

$$Y = X_{\text{centralizada}} \cdot Q$$

- Cada linha de Y representa um pixel projetado no espaço dos componentes principais.
- A transformação é reversível (sem perdas) se todos os componentes forem usados.

Redução de Dimensionalidade com Variância Explicada

A variância associada a cada componente é proporcional ao autovalor correspondente:

$$\text{Variância do PC}_i = \lambda_i$$

A variância total é $\Sigma\lambda_i$. Retendo apenas os k primeiros PCs que explicam 99% da variância total, garantimos alta fidelidade com menor dimensionalidade.

Propriedade de Ortogonalidade dos Autovetores

Autovetores distintos de uma matriz simétrica são ortogonais entre si.

- Isso garante que os componentes principais não sejam correlacionados entre si.
- Cada PC captura uma direção independente de variação nos dados.

Implementação

Antes da aplicação da Análise de Componentes Principais (PCA), foi necessário realizar uma etapa de preparação dos dados para garantir que a transformação fosse aplicada corretamente. Esta etapa envolveu a leitura, vetorização, centralização e empilhamento das bandas espectrais de uma imagem Landsat. As bandas espectrais (B1 à B9 - exceto B8) foram organizadas em arquivos .tif individuais.

A **implementação** percorre um pipeline claro: começa pela leitura das bandas Landsat, faz o empilhamento em um único raster, centraliza os dados e calcula a matriz de covariância; em seguida extrai os autovalores/vetores com SVD, projeta nos quatro primeiros componentes principais e, por fim, executa o K-Means em seis clusters antes de recortar o resultado pela máscara *.shp* e gerar as saídas *.tif/ .png* correspondentes README.

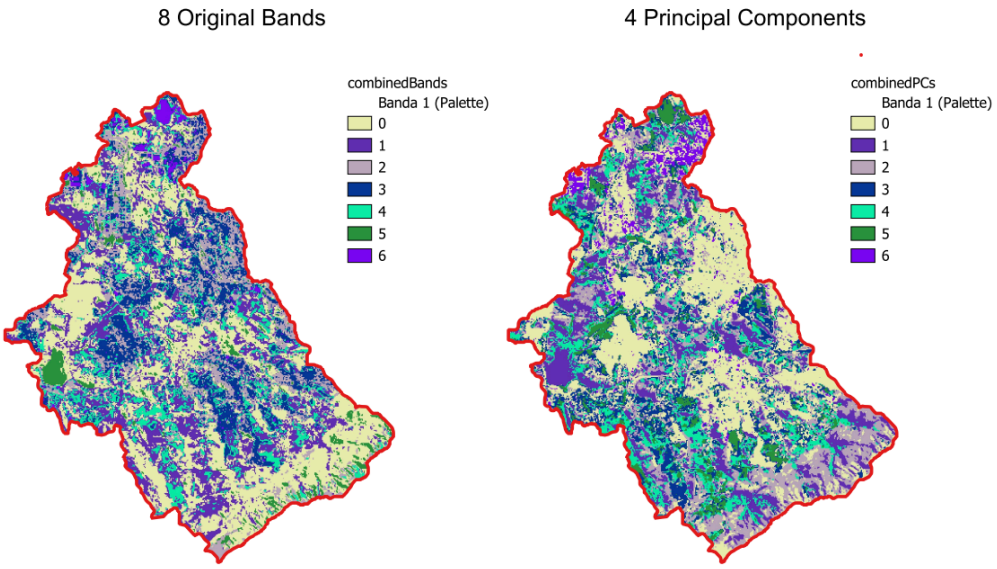
No repositório isso está organizado em módulos bem definidos: `gis.py` agrupa rotinas de I/O raster, clipping e o K-Means “snap-like”; `pca.py` contém a implementação própria da PCA via `numpy.linalg.svd`; `main.py` orquestra todo o fluxo. Scripts auxiliares (`config.bat`, `run.bat`) e pastas de dados (`Landsat Bands/`, `SHP_Bacia/`) completam a estrutura README.

Para reproduzir, clonar o projeto, rodar `config.bat` (prepara a *venv* e instala dependências) e depois `run.bat`, que executa automaticamente todas as etapas e salva os resultados em `data/Results/` README. O código está disponível em: https://github.com/ChrystianMelo/RemoteSensingPCA.

Resultados Obtidos

Área de estudo:

Resultados:



Observa-se que, quando o K-Means foi aplicado diretamente às 8 bandas originais, o mapa resultante exibiu uma granularização acentuada — pequenos fragmentos de classes espalhados que dificultam a interpretação espacial. Ao introduzir a PCA como etapa de pré-processamento, essa fragmentação diminuiu significativamente: as 4 componentes principais concentram a maior parte da variância relevante e filtram ruídos redundantes, permitindo que o algoritmo forme clusters maiores e mais coesos. Esse efeito é ainda mais notório nas zonas urbanas, onde a mistura espectral é alta; com a PCA, telhados, vias e edificações foram distinguidos de forma mais limpa das áreas de vegetação ou corpos d’água, demonstrando a vantagem do procedimento dimensionality-reduction para melhorar a clareza temática do mapeamento